ARMY RESEARCH LABORATORY

A VRML Door Prototype

by Andrew M. Neiderer

ARL-TR-3277 **August 2004**

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5067

ARL-TR-3277 **August 2004**

A VRML Door Prototype

Andrew M. Neiderer
Computational and Information Sciences Directorate, ARL

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
August 2004	Final	November 2003–April 2004

4. TITLE AND SUBTITLE
A VRML Door Prototype

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Andrew M. Neiderer

5d. PROJECT NUMBER
P622783.Y10

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: AMSRD-ARL-CI-CT
Aberdeen Proving Ground, MD 21005-5067

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-3277

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The objective of this research was to develop software that would contribute to military operations on urban terrain (MOUT) simulation. In particular, a virtual reality modeling language (VRML) 2.0 prototype for defining a door was completed. This VRML 2.0 node was also translated to the extensible 3D (X3D) language. An effort is now underway for describing apertures in general.

15. SUBJECT TERMS
VRML/X3D, MOUT, door PROTO

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Andrew M. Neiderer
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL	38	19b. TELEPHONE NUMBER (*Include area code*) 410-278-4999

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

1. Introduction

It is projected that by the year 2007 more than 50% of the world's population will live in urban areas.[1] This would be the first time in history that urban residents outnumber rural residents, and the trend is predicted to continue. Unfortunately, many demographers believe this will contribute to social problems within the urban environment. For example, a high influx of people could result in higher unemployment. This situation would be a strain on the local government to provide even the most basic services. In the worst case scenario, this could lead to social breakdown. Therefore peacekeeping, and in an extreme scenario, military operations on urban terrain (MOUT) may be necessary. With MOUT, any technological advantage is often compromised, e.g., a soldier's task may be reduced to forced entry of a window in a building instead of using some precision-guided weapon.

Since the 1980s, the U.S. military has been collaborating with the game entertainment industry when developing its computer simulators. Particularly, the U.S. Army recognized this convergence of combat simulation with gaming in the release of its first official video game, America's Army (AA), in the summer of 2002. AA is an online multiplayer game that is ideal for MOUT training. The Internet offers a solution for the delivery of data to participants. But it is the representation of this information that is our concern here.

The Virtual Reality Modeling Language (VRML) 2.0 offers a compelling presentation of three-dimensional (3-D) data with only minimal effort from the computer programmer. Its successor, called Extensible Three-Dimensional (X3D) graphics, was introduced in the summer of 2002. X3D uses the Extensible Markup Language (XML) technology and is more flexible in the Web environment. This may be the reason why it was recently selected as the scene graph language by the Moving Picture Experts Group (MPEG). Note that MPEG is the name of a family of standards used for coding audio-visual information.

Both VRML and X3D describe 3-D scenes in ASCII format. This promotes software development in an open environment, and should not be a limitation as hardware continues to obey Moore's law. (Moore's law predicts that processing power doubles every 18 months, while at the same time cost remains nearly constant.)

There is much published about the VRML standard, and little about X3D. X3D has just been standardized and may provide a better or more complete solution, but we are going to wait and see if this language becomes widespread like VRML.

[1] Department of Economic and Social Affairs. http://www.un.org/News/Press/docs/2004/pop899.doc.htm (accessed 24 March 2004).

This report describes in detail a VRML Door prototype node developed at the U.S. Army Research Laboratory (ARL) to assist in MOUT preparation. It is a VRML node that will be generalized for aperture description in the virtual world. The next section introduces VRML concepts necessary for understanding the node that we developed.[2] Section 2 describes our door prototype with details given in appendix A. Appendix B proposes an X3D translation of our prototype. We then conclude in section 4.

2. VRML 2.0 Basics

The VRML is a scene graph language originally designed for delivering 3-D content over the Internet. Probably the most predominant characteristic of the scene graph description is the parent/child relationship of VRML primitives, called nodes. As part of its definition, it includes the capability to specify geometry and its appearance, sense different conditions, and allow custom definition of actions. A VRML application is typically a hierarchical organization of these geometries, light sources, sensors, grouping primitives, and script methods assembled in a meaningful fashion. Note that the X3D graphics language appears to be the successor to VRML. It is an XML encoding of VRML with some additional functionality. Many believe that this language is destined to be the international standard for 3-D graphics content on the Web.

The VRML specification defines over 60 nodes (see figure 1), where each is a piece of information that implements some functionality. A node has an optional name and consists of a field interface (field, exposedField) and event interface (eventIn, eventOut). Each node generally falls into one of the following categories:

1. Basic Geometry node for simple shapes, while general shapes are built by listing individual coordinate values.
2. An *Appearance* node specifying the appearance attributes.
3. A *Shape* node which includes both the geometry and appearance.
4. A Light Source node to control lighting within the virtual world.
5. A Sensor node for detecting interactions between the user and elements of the visualization, or more specifically, other nodes.
6. Grouping nodes which allow a collection of objects to be defined and manipulated as one.
7. An Interpolator node for describing the changes that occur during an animation.

[2] Virtual Reality Modeling Language (VRML) International Standard ISO/IEC 14772. http://www.web3d.org/x3d/specifications/vrml/ISO_IEC_14772-All/index.html (accessed October 2003–April 2004).

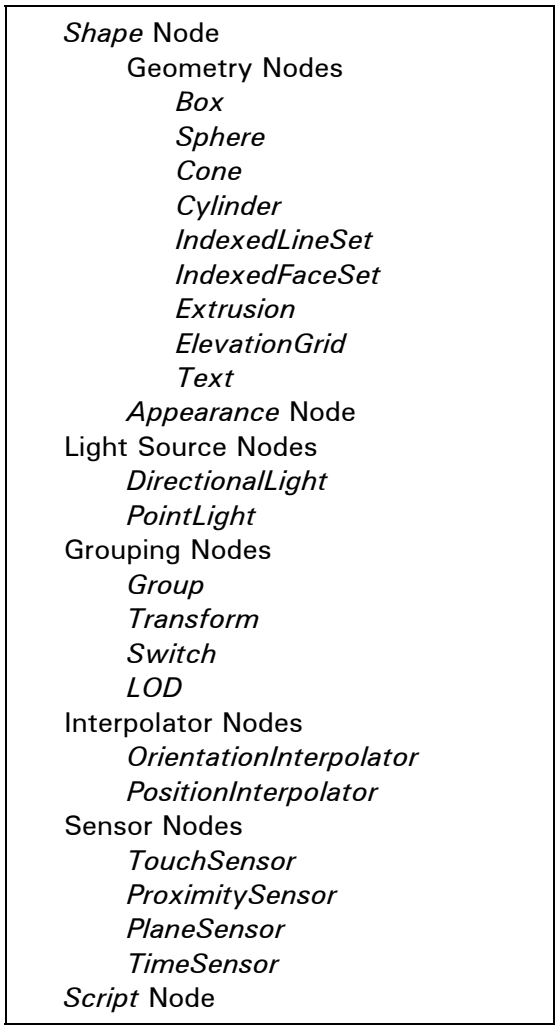

Figure 1. An abbreviated list of VRML 2.0 nodes.

8. A Script node for passing information between nodes by events whose routes are either explicitly declared or established.

Probably the most significant addition to the VRML 2.0 standard in December 1997 was better support for more dynamic display, change, and movement within a world. Animation is accomplished by defining a path for the flow of events among nodes. This is done by (1) using the ROUTE declaration to wire an output event from a sensor node to the input event of an interpolator node, and then (2) using ROUTE again, but this time from the interpolator node to a *Transform* node. VRML 2.0 also supports complex 3-D animations and behaviors by allowing Java and JavaScript programs to act upon VRML objects in a *Script* node.

Extensibility to the basic set of nodes is accomplished through a PROTO definition. That is, a prototype node enables the creation of a user-defined node. In section 3, a *Door* prototype node, which consists of many standard nodes, is defined for general use in a VRML environment.

The Cortona VRML Client 4.0 by ParallelGraphics, Inc. (www.parallelgraphics.com) was selected and installed for viewing our Web content. They have plug-ins for both Microsoft Internet Explorer and Netscape browsers. Others are available and can be reviewed at the Web3D Consortium Web site (www.web3d.org).

3. A VRML 2.0 Door

An urban environment has both static and dynamic components. For example, a building may be initially inactive until someone approaches and enters through one of its doors. It is the simple dynamics of door motion that we attempt to model here. Other elements must be similarly addressed if we are to simulate movement within the urban environment.

Although the VRML 2.0 language provides over 60 nodes for scene graph description of the virtual world being created, there are times when a user needs to define a custom node. A "new" node, or prototype, is nothing more than a combination of standard VRML nodes. The following discusses the field interface and nodes for our Door prototype.

Transform nodes are named for the Door itself, as well as its knob, lock plate, keyhole, and handle. Sensor (touch, time, and proximity) and interpolator (orientation) nodes are named for the distribution of events along appropriate paths when simulating the opening and closing of the door (see figure 2). The field interface of the Door node includes parameters for the following:

1. size – height, width, and thickness,
2. direction – "push" or "pull" direction of swing,
3. side – "left" or "right" side of opening,
4. sensorType – "proximity" or "touch" sensor for opening/closing,
5. proxSize – size of bounding box of ProximitySensor,
6. openingDuration and closingDuration – control timing for opening/closing,
7. audio – a Boolean value for sound effects,
8. doorTexture – an ImageTexture node,
9. doorMaterial – a Material node,
10. handleMaterial – a Material node, and
11. lockPlateMaterial – a Material node.

In addition, an AudioClip node has been added to increase realism for virtual simulation.

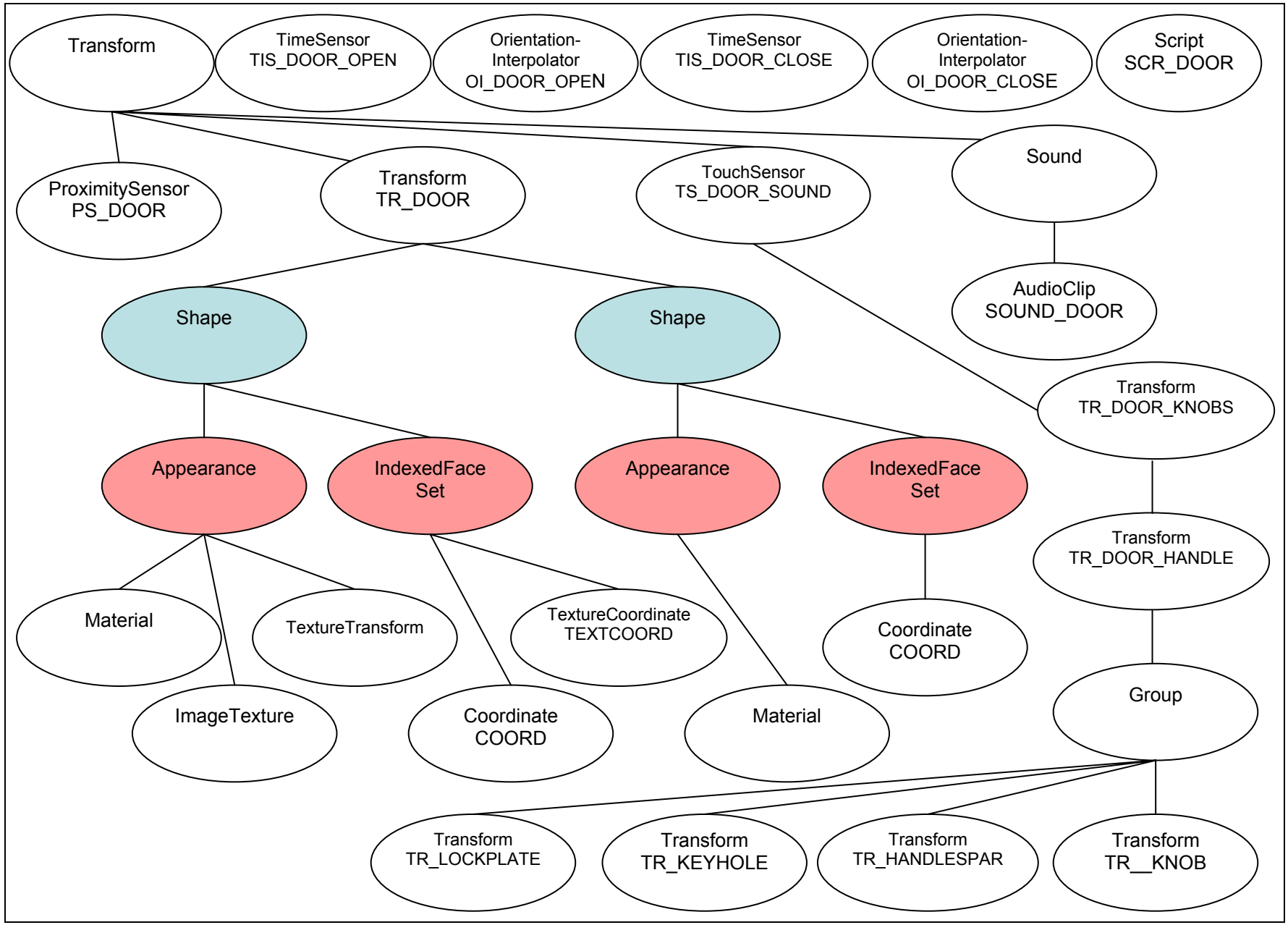

Figure 2. VRML nodes defining a Door prototype.

4. Conclusion

ARL has built a VRML door prototype node (see appendix A), and also the corresponding X3D component is proposed (see appendix B). VRML/X3D is being considered as the language for representation of data in future game development. There are nodes available for building interiors, e.g., opening/closing of windows and drawers of furniture (www.int3d.com). But some level of effort will be required to customize the structure of these nodes.

Appendix A. The Virtual Reality Modeling Language (VRML) 2.0 Door Prototype

The following VRML 2.0 code defines a prototype (PROTO) for the description of a door. The field interface gives some general characteristics of a door, and could be modified or tailored to a specific need. After the definition of the node is given, an external prototype declaration (EXTERNPROTO) is provided that makes use of this prototype.

The remainder of this appendix appears in its original form, without editorial change.

```
#VRML V2.0 utf8

# by Andrew M. Neiderer, 3 Mar 2004

EXTERNPROTO Door [

  # field interface

  field SFVec3f  size
  field SFString direction
  field SFString side
  field SFString sensorType
  field SFVec3f  proxSize
  field SFTime   openingDuration
  field SFTime   closingDuration
  field SFBool   audio

  field SFNode   doorTexture
  field SFNode   doorMaterial
  field SFNode   handleMaterial
  field SFNode   lockPlateMaterial
  # field MFString doorCreakURL ["creak5.wav"]
  field SFBool   doorEnabled

  # exposedField interface

  exposedField SFVec3f    translation
  exposedField SFRotation rotation
  exposedField SFColor    diffuseColor
  exposedField SFColor    specularColor
  exposedField SFColor    emissiveColor
  exposedField SFFloat    transparency
  exposedField MFString   textureUrl
  exposedField SFVec2f    textureScale

  # events interface

  eventIn      SFBool     doorHandleEnabled
] "Door.wrl#Door"

# outer door

Door {
  size              0.8 2.05 0.03
  translation       0.0 0.0  5.0
  side              "Left"
  direction         "Pull"
  sensorType        "Touch"
  textureUrl        "out_door.jpg"
  diffuseColor      0.2 0.5 0.5
  openingDuration   2.3
```

```
    closingDuration  2.3
    audio            TRUE
  }
```

```
#VRML V2.0 utf8

# a VRML door prototype by
#  Andrew M. Neiderer, 3 Mar 2004;
#  (original copyright Vincent Gardet - 2002).

# interface -

PROTO Door [

  # field interface

  field SFVec3f  size .85 2.05 .05        # size of the door
  field SFString direction "Push"         # direction of the opening : "Push" or "Pull"
  field SFString side "Right"             # side of the opening : "Left" or "Right"
  field SFString sensorType "Proximity"   # type of the sensor controlling the opening
                                          #  of the door : "Proximity" = ProximitySensor,
                                          #                "Touch" = TouchSensor
  field SFVec3f  proxSize 4.0 4.0 8.0     # size of the bounding box of the ProximitySensor,
  field SFTime   openingDuration 0.5
  field SFTime   closingDuration 1.0
  field SFBool   audio TRUE

  field SFNode   doorTexture ImageTexture {
                   url "wood.jpg"
                 }
  field SFNode   doorMaterial Material {
                   diffuseColor 0.4157 0.3019 0.1373
                 }
  field SFNode   handleMaterial Material {
                   diffuseColor 0.85 0.8 0.5
                   specularColor 0.9 0.9 0.8
                   shininess 0.9
                 }
  field SFNode   lockPlateMaterial Material {
                   diffuseColor 0.85 0.8 0.5
                   specularColor 0.9 0.9 0.8
                   shininess 0.0
                 }
  # field MFString doorCreakURL ["creak5.wav"]
  field SFBool   doorEnabled TRUE

  # exposedField interface

  exposedField   SFVec3f    translation 0.0 0.0 0.0
  exposedField   SFRotation rotation 0.0 0.0 0.0 0.0
  exposedField   SFColor    diffuseColor 0.8 0.8 0.8
  exposedField   SFColor    specularColor 0.0 0.0 0.0
  exposedField   SFColor    emissiveColor 0.0 0.0 0.0
  exposedField   SFFloat    transparency 0.0
  exposedField   MFString   textureUrl "" # if not symetrical, the texture of the
```

```
                                        #  door must present the handle at left side
    exposedField   SFVec2f textureScale 1.0 1.0

    exposedField   SFVec3f doorTranslation 0.0 0.0 0.0

    # events interface

    eventIn        SFBool doorHandleEnabled
  ]

  # definition -

  {
    Transform {
      translation  IS translation
      rotation     IS  rotation
      children [

       DEF PS_DOOR ProximitySensor {
         size IS proxSize
       }

       DEF TR_DOOR Transform {
         center .45 0 0
         children [
          Shape {
            appearance Appearance {
              material Material {
                diffuseColor  IS  diffuseColor
                specularColor IS  specularColor
                emissiveColor IS  emissiveColor
                transparency  IS  transparency
              }
              texture ImageTexture {
                url IS textureUrl
              }
              textureTransform TextureTransform {
                scale IS textureScale
              }
            } # end Appearance
            geometry IndexedFaceSet {
              coord DEF COORD Coordinate {}
              coordIndex [
               0,1,2,3,-1
               7,6,5,4,-1
              ]
              texCoord DEF TEXTCOORD TextureCoordinate {
                point [0 0,1 0,1 1,0 1]
              }
              texCoordIndex [
               0,1,2,3,-1
```

```
      3,2,1,0,-1
     ]
   } # end IndexedFaceSet
 } # end Shape
 Shape {
   appearance Appearance {
     material Material {
       diffuseColor  IS diffuseColor
       specularColor IS specularColor
       emissiveColor IS emissiveColor
       transparency  IS transparency
     }
   } # end Appearance
   geometry IndexedFaceSet {
     coord USE COORD
     coordIndex [
      1,5,6,2,-1
      4,0,3,7,-1
      2,6,7,3,-1
      0,4,5,1,-1
     ]
   } # end IndexedFaceSet
 } # end Shape

 # door knobs and lockplates

 DEF TR_DOOR_KNOBS Transform {
   translation 0.325 0.0 0.0
   rotation 0 1 0 0
   children [
    DEF TS_DOOR_KNOBS TouchSensor {}

    DEF TR_DOOR_HANDLE Transform {
      translation 0 0 0.0275
      children [
       Group {
         children [

          # lock plate

          DEF TR_LOCKPLATE Transform {
            translation 0 -0.05 0

            children [
             Shape {
               geometry Box {
                 size 0.1 0.25 0.01
               }
               appearance Appearance {
                 material IS lockPlateMaterial
               }
```

```
    } # end Shape
   ]
} # end TR_LOCKPLATE Transform

# key hole

DEF TR_KEYHOLE Transform {
  translation 0 -0.1 0.005
  children [
   Shape {
     geometry Box {
       size 0.02 0.05 0.001
     }
     appearance Appearance {
       material Material {
         diffuseColor 0 0 0
         shininess 0.0
       }
     }
   } # end Shape
  ]
} # end TR_KEYHOLE Transform

# handle spar

DEF TR_HANDLESPAR Transform {
  translation 0 0 0.03
  rotation 1 0 0 1.57

  children [
   Shape {
     geometry Cylinder {
       radius 0.015
       height 0.06
     }
     appearance Appearance {
       material IS handleMaterial
     }
   } # end Shape
  ]
} # end TR_HANDLESPAR Transform

# handle knob

DEF TR_DOORKNOB Transform {
  translation 0 0 0.07375
  children [
   Shape {
     geometry Sphere {
       radius 0.0275
     }
```

```
                    appearance Appearance {
                      material IS handleMaterial
                    }
                  } # end Shape

                  # add touch sensor to door knob

                  DEF TS_DOOR_HANDLE TouchSensor {
                    enabled IS doorHandleEnabled
                  }
                ]
              } # end TR_DOORKNOB Transform
            ]
          } # end Group
        ]
      } # end TR_DOOR_HANDLE Transform

      # opposite side of door

      Transform {
        translation 0 0 0
        rotation 0 1 0 3.14
        children USE TR_DOOR_HANDLE
      }

      DEF TS_DOOR_SOUND TouchSensor {}
      Sound {
        source DEF SOUND_DOOR AudioClip {
          url "DOOR3.WAV"
          loop FALSE
        }
      }
    ]
  } # end TR_DOOR_KNOBS Transform
 ]
 } # end TR_DOOR
 ]
} # end Transform

DEF TIS_OPEN_DOOR TimeSensor {
  cycleInterval IS openingDuration
  loop FALSE
}

DEF  OI_OPEN_DOOR OrientationInterpolator {
  key      [ 0.0, 1.0 ]
  keyValue [ 0.0 1.0 0.0 0.0, 0.0 1.0 0.0 -1.5708 ]
}

DEF TIS_CLOSE_DOOR TimeSensor {
  cycleInterval IS closingDuration
```

```
  loop FALSE
}

DEF  OI_CLOSE_DOOR OrientationInterpolator {
  key       [ 0.0, 1.0 ]
  keyValue [ 0.0 1.0 0.0 -1.5708, 0.0 1.0 0.0 0.0 ]
}

DEF SCR_B_DOOR Script {
  directOutput TRUE

  # field interface

  field SFVec3f size IS size
  field SFString sensorType IS sensorType
  field SFString side IS side
  field SFString direction IS direction
  field SFBool state FALSE
  field SFNode OI_OPEN_DOOR USE OI_OPEN_DOOR
  field SFNode OI_CLOSE_DOOR USE OI_CLOSE_DOOR
  field SFNode COORD USE COORD
  field SFNode TEXTCOORD USE TEXTCOORD

  # eventIn, eventOut interface

  eventIn  SFTime  touchDoor
  eventIn  SFTime  enterDoor
  eventIn  SFTime  exitDoor
  eventOut SFBool  Pstate
  eventOut SFBool  Tstate
  eventOut SFTime  openDoor
  eventOut SFTime  closeDoor
  eventOut SFVec3f turningPoint

  eventIn  SFTime  TS_DOOR_HANDLE

  url "javascript:
   function initialize() {

     COORD.point[0][0] = -size.x / 2;
     COORD.point[0][1] = -size.y / 2;
     COORD.point[0][2] = size.z / 2;
     COORD.point[1][0] = size.x / 2;
     COORD.point[1][1] = -size.y / 2;
     COORD.point[1][2] = size.z / 2;
     COORD.point[2][0] = size.x / 2;
     COORD.point[2][1] = size.y / 2;
     COORD.point[2][2] = size.z / 2;
     COORD.point[3][0] = -size.x / 2;
     COORD.point[3][1] = size.y / 2;
     COORD.point[3][2] = size.z / 2;
```

```
COORD.point[4][0] = -size.x / 2;
COORD.point[4][1] = -size.y / 2;
COORD.point[4][2] = -size.z / 2;
COORD.point[5][0] = size.x / 2;
COORD.point[5][1] = -size.y / 2;
COORD.point[5][2] = -size.z / 2;
COORD.point[6][0] = size.x / 2;
COORD.point[6][1] = size.y / 2;
COORD.point[6][2] = -size.z / 2;
COORD.point[7][0] = -size.x / 2;
COORD.point[7][1] = size.y / 2;
COORD.point[7][2] = -size.z / 2;

turningPoint[0] = ((size.x / 2) - (size.z / 2));
turningPoint[1] = 0;
turningPoint[2] = 0;

if ( sensorType == 'Touch' ) {
  Pstate = false;
}
else {
  Tstate = false;
}

if ( side == 'Left' ) {
  turningPoint[0] = -((size.x / 2) - (size.z / 2));
  turningPoint[1] = 0;
  turningPoint[2] = 0;

  TEXTCOORD.point[0][0] = 1;
  TEXTCOORD.point[0][1] = 0;
  TEXTCOORD.point[1][0] = 0;
  TEXTCOORD.point[1][1] = 0;
  TEXTCOORD.point[2][0] = 0;
  TEXTCOORD.point[2][1] = 1;
  TEXTCOORD.point[3][0] = 1;
  TEXTCOORD.point[3][1] = 1;
}

if ( ((side == 'Left')  && (direction == 'Push')) ||
     ((side == 'Right') && (direction == 'Pull')) ) {

  OI_OPEN_DOOR.keyValue[0][0] = 0;
  OI_OPEN_DOOR.keyValue[0][1] = 1;
  OI_OPEN_DOOR.keyValue[0][2] = 0;
  OI_OPEN_DOOR.keyValue[0][3] = 0;
  OI_OPEN_DOOR.keyValue[1][0] = 0;
  OI_OPEN_DOOR.keyValue[1][1] = 1;
  OI_OPEN_DOOR.keyValue[1][2] = 0;
  OI_OPEN_DOOR.keyValue[1][3] = 1.5708;
```

```
        OI_CLOSE_DOOR.keyValue[0][0] = 0;
        OI_CLOSE_DOOR.keyValue[0][1] = 1;
        OI_CLOSE_DOOR.keyValue[0][2] = 0;
        OI_CLOSE_DOOR.keyValue[0][3] = 1.5708;
        OI_CLOSE_DOOR.keyValue[1][0] = 0;
        OI_CLOSE_DOOR.keyValue[1][1] = 1;
        OI_CLOSE_DOOR.keyValue[1][2] = 0;
        OI_CLOSE_DOOR.keyValue[1][3] = 0;
      }
    }

    function touchDoor (value) {

      if ( state == FALSE ) {
        state = TRUE;
        openDoor = value;
      }
      else {
        state = FALSE;
        closeDoor = value;
      }
    }

    function enterDoor (value) {

      if ( state == FALSE ) {
        state = TRUE;
        openDoor = value;
      }
    }

    function exitDoor (value) {

      if ( state == TRUE ) {
        state = FALSE;
        closeDoor = value;
      }
    }
   "
}

# animation

ROUTE SCR_B_DOOR.Tstate TO TS_DOOR_KNOBS.set_enabled
ROUTE SCR_B_DOOR.Pstate TO PS_DOOR.set_enabled

ROUTE SCR_B_DOOR.turningPoint TO  TR_DOOR.set_center

ROUTE TS_DOOR_KNOBS.touchTime TO SCR_B_DOOR.touchDoor
ROUTE PS_DOOR.enterTime TO SCR_B_DOOR.enterDoor
ROUTE PS_DOOR.exitTime TO SCR_B_DOOR.exitDoor
```

```
    ROUTE SCR_B_DOOR.openDoor TO TIS_OPEN_DOOR.startTime
    ROUTE TIS_OPEN_DOOR.fraction_changed TO OI_OPEN_DOOR.set_fraction
    ROUTE OI_OPEN_DOOR.value_changed TO TR_DOOR.set_rotation

    ROUTE SCR_B_DOOR.closeDoor TO TIS_CLOSE_DOOR.startTime
    ROUTE TIS_CLOSE_DOOR.fraction_changed TO OI_CLOSE_DOOR.set_fraction
    ROUTE OI_CLOSE_DOOR.value_changed TO TR_DOOR.set_rotation

    ROUTE TS_DOOR_SOUND.touchTime TO SOUND_DOOR.set_startTime
}
```

Appendix B. The Extensible Three-Dimensional (X3D) Door Prototype

This appendix provides an X3D translation of our door prototype. In general, Virtual Reality Modeling Language (VRML) 2.0 nodes are now tags with attributes, e.g., <Material shininess=”0.2” diffuseColor=”0.3 0.6 0.1”>, which is consistent with Extensible Markup Language (XML) formatting.

The remainder of this appendix appears in its original form, without editorial change.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC
 "http://www.web3D.org/TaskGroups/x3d/translation/x3d-compact.dtd"
 "/www.web3D.org/TaskGroups/x3d/translation/x3d-compact.dtd">

<X3D>

  <head>
    <meta name="filename" content="doors.x3d"/>
    <meta name="description"
          content="*enter description here, short-sentence summaries preferred*"/>
    <meta name="author" content="*enter name of original author here*"/>
    <meta name="translator"
          content="*if manually translating VRML-to-X3D, enter name of person translating here*"/>
    <meta name="created" content="*enter date of initial version here*"/>
    <meta name="revised" content="*enter date of latest revision here*"/>
    <meta name="version" content="*enter version here*"/>
    <meta name="reference"
          content="*enter reference citation or relative/online url here*"/>
    <meta name="reference"
          content="*enter additional url/bibliographic reference information here*"/>
    <meta name="copyright"
          content="*enter copyright information here* Example:  Copyright (c) Web3D Consortium Inc. 2001"/>
    <meta name="drawing"
          content="*enter drawing filename/url here*"/>
    <meta name="image" content="*enter image filename/url here*"/>
    <meta name="movie" content="*enter movie filename/url here*"/>
    <meta name="photo" content="*enter photo filename/url here*"/>
    <meta name="keywords" content="*enter keywords here*"/>
    <meta name="url"
          content="*enter online url address for this file here*"/>
    <meta name="generator"
          content="Vrml97ToX3dNist, http://ovrt.nist.gov/v2_x3d.html"/>
  </head>

  <Scene>
    <ExternProtoDeclare name="Door"  url='"Door.wrl#Door"'>
      <field  name="proxSize" type="Vector3Float" accessType="field"/>
      <field  name="side" type="String" accessType="field"/>
      <field  name="doorTexture" type="Node" accessType="field"/>
      <field  name="textureUrl" type="Strings" accessType="exposedField"/>
      <field  name="textureScale" type="Vector2Float" accessType="exposedField"/>
      <field  name="translation" type="Vector3Float" accessType="exposedField"/>
      <field  name="specularColor" type="Color" accessType="exposedField"/>
      <field  name="sensorType" type="String" accessType="field"/>
      <field  name="diffuseColor" type="Color" accessType="exposedField"/>
      <field  name="size" type="Vector3Float" accessType="field"/>
      <field  name="doorHandleEnabled" type="Boolean" accessType="eventIn"/>
      <field  name="lockPlateMaterial" type="Node" accessType="field"/>
      <field  name="transparency" type="Float" accessType="exposedField"/>
      <field  name="closingDuration" type="Time" accessType="field"/>
```

```
      <field  name="rotation" type="Rotation" accessType="exposedField"/>
      <field  name="doorMaterial" type="Node" accessType="field"/>
      <field  name="openingDuration" type="Time" accessType="field"/>
      <field  name="emissiveColor" type="Color" accessType="exposedField"/>
      <field  name="handleMaterial" type="Node" accessType="field"/>
      <field  name="direction" type="String" accessType="field"/>
      <field  name="audio" type="Boolean" accessType="field"/>
      <field  name="doorEnabled" type="Boolean" accessType="field"/>
    </ExternProtoDeclare>

    <ProtoInstance name="Door">
      <fieldValue name="side" value="Left"/>
      <fieldValue name="textureUrl" value='"out_door.jpg"'/>
      <fieldValue name="translation" value="0.0 0.0 5.0"/>
      <fieldValue name="sensorType" value="Touch"/>
      <fieldValue name="diffuseColor" value="0.2 0.5 0.5"/>
      <fieldValue name="size" value="0.8 2.05 0.03"/>
      <fieldValue name="closingDuration" value="2.3"/>
      <fieldValue name="openingDuration" value="2.3"/>
      <fieldValue name="direction" value="Pull"/>
      <fieldValue name="audio" value="true"/>
    </ProtoInstance>
  </Scene>

</X3D>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC
 "http://www.web3D.org/TaskGroups/x3d/translation/x3d-compact.dtd"
 "/www.web3D.org/TaskGroups/x3d/translation/x3d-compact.dtd">

<X3D>

  <head>
    <meta name="filename" content="Door.x3d"/>
    <meta name="description"
          content="*enter description here, short-sentence summaries preferred*"/>
    <meta name="author"
          content="*enter name of original author here*"/>
    <meta name="translator"
          content="*if manually translating VRML-to-X3D, enter name of person translating here*"/>
    <meta name="created" content="*enter date of initial version here*"/>
    <meta name="revised" content="*enter date of latest revision here*"/>
    <meta name="version" content="*enter version here*"/>
    <meta name="reference"
          content="*enter reference citation or relative/online url here*"/>
    <meta name="reference"
          content="*enter additional url/bibliographic reference information here*"/>
    <meta name="copyright"
          content="*enter copyright information here* Example:  Copyright (c) Web3D Consortium Inc. 2001"/>
    <meta name="drawing" content="*enter drawing filename/url here*"/>
    <meta name="image" content="*enter image filename/url here*"/>
    <meta name="movie" content="*enter movie filename/url here*"/>
    <meta name="photo" content="*enter photo filename/url here*"/>
    <meta name="keywords" content="*enter keywords here*"/>
    <meta name="url" content="*enter online url address for this file here*"/>
    <meta name="generator"
          content="Vrml97ToX3dNist, http://ovrt.nist.gov/v2_x3d.html"/>
  </head>

  <Scene>
    <ProtoDeclare name="Door">
      <field IS="PS_DOOR.size" name="proxSize" type="Vector3Float"
             value="4.0 4.0 8.0" accessType="field"/>
      <field IS="SCR_B_DOOR.side" name="side" type="String"
             value="Right" accessType="field"/>
      <field name="doorTexture" type="Node" accessType="field">
        <ImageTexture url='"wood.jpg"'/>
      </field>
      <field IS="_IS_2.url" name="textureUrl" type="Strings"
             value='""' accessType="exposedField"/>
      <field IS="_IS_3.scale" name="textureScale" type="Vector2Float"
             value="1.0 1.0" accessType="exposedField"/>
      <field IS="_IS_0.translation" name="translation" type="Vector3Float"
             value="0.0 0.0 0.0" accessType="exposedField"/>
      <field IS="_IS_1.specularColor _IS_4.specularColor" name="specularColor"
             type="Color" value="0.0 0.0 0.0" accessType="exposedField"/>
```

```
<field IS="SCR_B_DOOR.sensorType" name="sensorType" type="String"
       value="Proximity" accessType="field"/>
<field IS="_IS_1.diffuseColor _IS_4.diffuseColor" name="diffuseColor"
       type="Color" value="0.8 0.8 0.8" accessType="exposedField"/>
<field IS="SCR_B_DOOR.size" name="size" type="Vector3Float"
       value="0.85 2.05 0.05" accessType="field"/>
<field IS="touchHandle.enabled" name="doorHandleEnabled" type="Boolean"
       accessType="eventIn"/>
<field IS="_IS_5.material" name="lockPlateMaterial" type="Node"
       accessType="field">
  <Material shininess="0.0" diffuseColor="0.85 0.8 0.5"
            specularColor="0.9 0.9 0.8"/>
</field>
<field IS="_IS_1.transparency _IS_4.transparency" name="transparency"
       type="Float" value="0.0" accessType="exposedField"/>
<field IS="TIS_CLOSE_DOOR.cycleInterval" name="closingDuration"
       type="Time" value="1.0" accessType="field"/>
<field IS="_IS_0.rotation" name="rotation" type="Rotation"
       value="0.0 0.0 0.0  0.0" accessType="exposedField"/>
<field name="doorMaterial" type="Node" accessType="field">
  <Material diffuseColor="0.4157 0.3019 0.1373"/>
</field>
<field IS="TIS_OPEN_DOOR.cycleInterval" name="openingDuration"
       type="Time" value="0.5" accessType="field"/>
<field name="doorTranslation" type="Vector3Float" value="0.0 0.0 0.0"
       accessType="exposedField"/>
<field IS="_IS_1.emissiveColor _IS_4.emissiveColor" name="emissiveColor"
     type="Color" value="0.0 0.0 0.0" accessType="exposedField"/>
<field IS="_IS_6.material _IS_7.material" name="handleMaterial"
       type="Node" accessType="field">
  <Material shininess="0.9" diffuseColor="0.85 0.8 0.5"
            specularColor="0.9 0.9 0.8"/>
</field>
<field IS="SCR_B_DOOR.direction" name="direction" type="String"
       value="Push" accessType="field"/>
<field name="audio" type="Boolean" value="true" accessType="field"/>
<field name="doorEnabled" type="Boolean" value="true" accessType="field"/>

<Transform DEF="_IS_0">
  <ProximitySensor DEF="PS_DOOR"/>
  <Transform DEF="TR_DOOR" center="0.45 0.0 0.0">
    <Shape>
      <Appearance >
        <TextureTransform DEF="_IS_3"/>
        <Material DEF="_IS_1"/>
        <ImageTexture DEF="_IS_2"/>
      </Appearance>
      <IndexedFaceSet coordIndex=" 0 1 2 3 -1 7 6 5 4 -1"
                      texCoordIndex=" 0 1 2 3 -1 3 2 1 0 -1">
        <TextureCoordinate DEF="TEXTCOORD"
                           point="0.0 0.0, 1.0 0.0, 1.0 1.0, 0.0 1.0"/>
```

```
      <Coordinate DEF="COORD"/>
    </IndexedFaceSet>
  </Shape>
  <Shape>
    <Appearance>
      <Material DEF="_IS_4"/>
    </Appearance>
    <IndexedFaceSet coordIndex=" 1 5 6 2 -1 4 0 3 7 -1 2 6 7 3 -1 0 4 5 1 -1">
      <Coordinate USE="COORD"/>
    </IndexedFaceSet>
  </Shape>
  <Transform DEF="Door_Knobs" translation="0.325 0.0 0.0"
             rotation="0.0 1.0 0.0  0.0">
    <TouchSensor DEF="TS_Door_Knobs"/>
    <Transform DEF="Door_Handle" translation="0.0 0.0 0.0275">
      <Group>
        <Transform translation="0.0 -0.05 0.0">
          <Shape>
            <Appearance DEF="_IS_5">
            </Appearance>
            <Box size="0.1 0.25 0.01"/>
          </Shape>
        </Transform>
        <Transform translation="0.0 -0.1 0.0050">
          <Shape>
            <Appearance>
              <Material shininess="0.0" diffuseColor="0.0 0.0 0.0"/>
            </Appearance>
            <Box size="0.02 0.05 0.0010"/>
          </Shape>
        </Transform>
        <Transform translation="0.0 0.0 0.03" rotation="1.0 0.0 0.0  1.57">
          <Shape>
            <Appearance DEF="_IS_6">
            </Appearance>
            <Cylinder radius="0.015" height="0.06"/>
          </Shape>
        </Transform>
        <Transform translation="0.0 0.0 0.07375">
          <Shape>
            <Appearance DEF="_IS_7">
            </Appearance>
            <Sphere radius="0.0275"/>
          </Shape>
          <TouchSensor DEF="touchHandle"/>
        </Transform>
      </Group>
    </Transform>
    <Transform translation="0.0 0.0 0.0"
               rotation="0.0 1.0 0.0  3.14">
      <Transform USE="Door_Handle"/>
```

```
        </Transform>
        <TouchSensor DEF="TS_SOUND_DOOR"/>
        <Sound>
          <AudioClip DEF="SOUND_DOOR" loop="false" url='"DOOR3.WAV"'/>
        </Sound>
      </Transform>
    </Transform>
  </Transform>

  <TimeSensor DEF="TIS_OPEN_DOOR" loop="false"/>
  <OrientationInterpolator DEF="OI_OPEN_DOOR" key=" 0.0 1.0" keyValue="0.0 1.0 0.0  0.0, 0.0 1.0 0.0  -1.5708, "/>
  <TimeSensor DEF="TIS_CLOSE_DOOR" loop="false"/>
  <OrientationInterpolator DEF="OI_CLOSE_DOOR" key=" 0.0 1.0" keyValue="0.0 1.0 0.0  -1.5708, 0.0 1.0 0.0  0.0, "/>

  <Script xml:space="preserve" DEF="SCR_B_DOOR" directOutput="true">
    <![CDATA[
    javascript:
    function initialize() {

      COORD.point[0][0] = -size.x / 2;
      COORD.point[0][1] = -size.y / 2;
      COORD.point[0][2] = size.z / 2;
      COORD.point[1][0] = size.x / 2;
      COORD.point[1][1] = -size.y / 2;
      COORD.point[1][2] = size.z / 2;
      COORD.point[2][0] = size.x / 2;
      COORD.point[2][1] = size.y / 2;
      COORD.point[2][2] = size.z / 2;
      COORD.point[3][0] = -size.x / 2;
      COORD.point[3][1] = size.y / 2;
      COORD.point[3][2] = size.z / 2;
      COORD.point[4][0] = -size.x / 2;
      COORD.point[4][1] = -size.y / 2;
      COORD.point[4][2] = -size.z / 2;
      COORD.point[5][0] = size.x / 2;
      COORD.point[5][1] = -size.y / 2;
      COORD.point[5][2] = -size.z / 2;
      COORD.point[6][0] = size.x / 2;
      COORD.point[6][1] = size.y / 2;
      COORD.point[6][2] = -size.z / 2;
      COORD.point[7][0] = -size.x / 2;
      COORD.point[7][1] = size.y / 2;
      COORD.point[7][2] = -size.z / 2;

      turningPoint[0] = ((size.x / 2) - (size.z / 2));
      turningPoint[1] = 0;
      turningPoint[2] = 0;

      if ( sensorType == 'Touch' ) {
        Pstate = false;
      }
```

```
    else {
      Tstate = false;
    }

    if ( side == 'Left' ) {
      turningPoint[0] = -((size.x / 2) - (size.z / 2));
      turningPoint[1] = 0;
      turningPoint[2] = 0;

      TEXTCOORD.point[0][0] = 1;
      TEXTCOORD.point[0][1] = 0;
      TEXTCOORD.point[1][0] = 0;
      TEXTCOORD.point[1][1] = 0;
      TEXTCOORD.point[2][0] = 0;
      TEXTCOORD.point[2][1] = 1;
      TEXTCOORD.point[3][0] = 1;
      TEXTCOORD.point[3][1] = 1;
    }

    if ( ((side == 'Left')  && (direction == 'Push')) ||
         ((side == 'Right') && (direction == 'Pull')) ) {

      OI_OPEN_DOOR.keyValue[0][0] = 0;
      OI_OPEN_DOOR.keyValue[0][1] = 1;
      OI_OPEN_DOOR.keyValue[0][2] = 0;
      OI_OPEN_DOOR.keyValue[0][3] = 0;
      OI_OPEN_DOOR.keyValue[1][0] = 0;
      OI_OPEN_DOOR.keyValue[1][1] = 1;
      OI_OPEN_DOOR.keyValue[1][2] = 0;
      OI_OPEN_DOOR.keyValue[1][3] = 1.5708;

      OI_CLOSE_DOOR.keyValue[0][0] = 0;
      OI_CLOSE_DOOR.keyValue[0][1] = 1;
      OI_CLOSE_DOOR.keyValue[0][2] = 0;
      OI_CLOSE_DOOR.keyValue[0][3] = 1.5708;
      OI_CLOSE_DOOR.keyValue[1][0] = 0;
      OI_CLOSE_DOOR.keyValue[1][1] = 1;
      OI_CLOSE_DOOR.keyValue[1][2] = 0;
      OI_CLOSE_DOOR.keyValue[1][3] = 0;
    }
  }

  function touchDoor (value) {

    if ( state == FALSE ) {
      state = TRUE;
      openDoor = value;
    }
    else {
      state = FALSE;
      closeDoor = value;
```

```
  }
}

function enterDoor (value) {

  if ( state == FALSE ) {
    state = TRUE;
    openDoor = value;
  }
}

function exitDoor (value) {

  if ( state == TRUE ) {
    state = FALSE;
    closeDoor = value;
  }
}

]]>

<field xml:space="preserve" name="enterDoor" type="Time"
       accessType="eventIn"/>
<field xml:space="preserve" name="side" type="String" value=""
       accessType="field"/>
<field xml:space="preserve" name="closeDoor" type="Time"
       accessType="eventOut"/>
<field xml:space="preserve" name="state" type="Boolean" value="false"
       accessType="field"/>
<field xml:space="preserve" name="COORD" type="Node" USE="COORD"
       accessType="field"/>
<field xml:space="preserve" name="Tstate" type="Boolean"
       accessType="eventOut"/>
<field xml:space="preserve" name="exitDoor" type="Time"
       accessType="eventIn"/>
<field xml:space="preserve" name="OI_CLOSE_DOOR" type="Node"
       USE="OI_CLOSE_DOOR" accessType="field"/>
<field xml:space="preserve" name="sensorType" type="String"
       value="" accessType="field"/>
<field xml:space="preserve" name="size" type="Vector3Float"
       value="0.0 0.0 0.0" accessType="field"/>
<field xml:space="preserve" name="TEXTCOORD" type="Node"
       USE="TEXTCOORD" accessType="field"/>
<field xml:space="preserve" name="turningPoint" type="Vector3Float"
       accessType="eventOut"/>
<field xml:space="preserve" name="touchDoor" type="Time"
       accessType="eventIn"/>
<field xml:space="preserve" name="openDoor" type="Time"
       accessType="eventOut"/>
<field xml:space="preserve" name="Pstate" type="Boolean"
       accessType="eventOut"/>
```

```
            <field xml:space="preserve" name="OI_OPEN_DOOR" type="Node"
                   USE="OI_OPEN_DOOR" accessType="field"/>
            <field xml:space="preserve" name="direction" type="String"
                   value="" accessType="field"/>
            <field xml:space="preserve" name="touchHandle" type="Time"
                   accessType="eventIn"/>
          </Script>

          <ROUTE fromNode="SCR_B_DOOR" fromField="Tstate" toNode="TS_Door_Knobs"
                 toField="set_enabled"/>
          <ROUTE fromNode="SCR_B_DOOR" fromField="Pstate" toNode="PS_DOOR"
                 toField="set_enabled"/>
          <ROUTE fromNode="SCR_B_DOOR" fromField="turningPoint" toNode="TR_DOOR"
                 toField="set_center"/>
          <ROUTE fromNode="TS_Door_Knobs" fromField="touchTime" toNode="SCR_B_DOOR"
                 toField="touchDoor"/>
          <ROUTE fromNode="PS_DOOR" fromField="enterTime" toNode="SCR_B_DOOR"
                 toField="enterDoor"/>
          <ROUTE fromNode="PS_DOOR" fromField="exitTime" toNode="SCR_B_DOOR"
                 toField="exitDoor"/>
          <ROUTE fromNode="SCR_B_DOOR" fromField="openDoor" toNode="TIS_OPEN_DOOR"
                 toField="startTime"/>
          <ROUTE fromNode="TIS_OPEN_DOOR" fromField="fraction_changed" toNode="OI_OPEN_DOOR"
                 toField="set_fraction"/>
          <ROUTE fromNode="OI_OPEN_DOOR" fromField="value_changed" toNode="TR_DOOR"
                 toField="set_rotation"/>
          <ROUTE fromNode="SCR_B_DOOR" fromField="closeDoor" toNode="TIS_CLOSE_DOOR"
                 toField="startTime"/>
          <ROUTE fromNode="TIS_CLOSE_DOOR" fromField="fraction_changed" toNode="OI_CLOSE_DOOR"
                 toField="set_fraction"/>
          <ROUTE fromNode="OI_CLOSE_DOOR" fromField="value_changed" toNode="TR_DOOR"
                 toField="set_rotation"/>
          <ROUTE fromNode="TS_SOUND_DOOR" fromField="touchTime" toNode="SOUND_DOOR"
                 toField="set_startTime"/>
        </ProtoDeclare>
      </Scene>

    </X3D>
```

Bibliography

Ames, A. L.; Nadeau, D. R.; Moreland, J. L. VRML 2.0 Sourcebook; John Wiley & Sons, Inc.: New York, NY, 1997.

The web3D Consortium. Extensible 3D (X3D) International Draft Standards. http://www.web3d.org/x3d/specifications.

The web3D Consortium. The Virtual Reality Modeling Language. http://www.web3d.org/x3d/specifications/vrml/ISO_IEC_14772-All/index.html.

NO. OF COPIES	ORGANIZATION
1 (PDF Only)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218
1	COMMANDING GENERAL US ARMY MATERIEL CMD AMCRDA TF 5001 EISENHOWER AVE ALEXANDRIA VA 22333-0001
1	INST FOR ADVNCD TCHNLGY THE UNIV OF TEXAS AT AUSTIN 3925 W BRAKER LN STE 400 AUSTIN TX 78759-5316
1	US MILITARY ACADEMY MATH SCI CTR EXCELLENCE MADN MATH THAYER HALL WEST POINT NY 10996-1786
1	DIRECTOR US ARMY RESEARCH LAB AMSRD ARL CS IS R 2800 POWDER MILL RD ADELPHI MD 20783-1197
3	DIRECTOR US ARMY RESEARCH LAB AMSRD ARL CI OK TL 2800 POWDER MILL RD ADELPHI MD 20783-1197
3	DIRECTOR US ARMY RESEARCH LAB AMSRD ARL CS IS T 2800 POWDER MILL RD ADELPHI MD 20783-1197

NO. OF COPIES	ORGANIZATION
	ABERDEEN PROVING GROUND
1	DIR USARL AMSRD ARL CI OK TP (BLDG 4600)

NO. OF
COPIES ORGANIZATION

ABERDEEN PROVING GROUND

7 DIR USARL
AMSRD ARL CI CT
P JONES
M THOMAS
A NEIDERER (5 CPS)

INTENTIONALLY LEFT BLANK.